February 7, 2001

DIGITAL CIRCUIT SIMULATION USING HSPICE

Charles R. Kime
Dept. of Electrical and Computer Engineering
University of Wisconsin – Madison

The pdf version of this document has extensive hyperlinks shown in blue.

TABLE OF CONTENTS

GETTING STARTED WITH HSPICE 2
- The HSPICE Netlist File 2
- An Example HSPICE File 3
- Executing HSPICE 4
- Displaying HSPICE Output 6
- Output Printing 6
- HSPICE User's Manual and AvanWaves Help and Manual 7

HSPICE COMMANDS 7
INPUT WAVEFORMS 9
OUTPUT VARIABLES 10
EXPRESSIONS 10
SPECIFIC ANALYSES AND OUTPUT 12
- I-V Curves 12
- Family of I-V Curves 14
- I-V Curve for Parameter Extraction 15
- Voltage Transfer Characteristic (VTC) 17
- Transient (Delay) Analysis 17

CONVERGENCE PROBLEMS 19
- DC Analysis 19
- Transient Analysis 19
- Diode Convergence 19

EXECUTING HSPICE FROM THE COMMAND LINE 19

REFERENCES 19
FEEDBACK 20

GETTING STARTED WITH HSPICE

Avant!'s Star-Hspice (referred to here as HSPICE) is available on the HP Unix workstations at CAE. HSPICE executes in batch mode using the Unix command line or a graphical user interface called AvanWaves. Here we will emphasize the use of AvanWaves on HP Unix workstations, with a brief section at the end on use of command line execution of HSPICE.

HSPICE uses a netlist file `design.sp`, where `design` is the name of your circuit, as a source file. This text file contains the circuit netlist, element models, analysis commands and output commands. Execution of HSPICE produces a number of files depending on user-specified options. By use of the appropriate options, files are produced which act as the input files for AvanWaves for displaying, analyzing, and printing results from HSPICE.

In addition, text files for reading and printing and graphic files for direct printing from HSPICE can be produced.

The HSPICE Netlist File

Although some versions of SPICE required the use of upper case letters, you may also use lower case letters with this version. Further, this version is not case sensitive, so `Vin` and `vin` are recognized as the same variable! Lines containing comments begin with *. Comments within other lines begin with one or more spaces followed by $. To continue a statement on multiple lines, each line after the first must have a + in the first column, or each line of the statement except for the last line must end with \ or \\. \\ removes all spaces. For example,

```
ABC \
DEF
```

gives ABC DEF. But,

```
ABC \\
DEF
```

gives ABCDEF.

An Example HSPICE File

An NMOS depletion-mode load inverter illustrates the components of a typical digital circuit HSPICE file. The numbers appearing on the right of the file are **not** part of the file but appear only for reference.

```
NMOS Depletion-Mode Inverter (file: nmos_inv.sp)          1
*Last revised 1/11/97                                     2
*Power Supplies                                           3
Vdd 1 0 DC +5V                                            4
```

```
Vbb 2 0 DC -2V                                                    5
*Input Signal                                                     6
*Vin 3 0 PWL 0ns 0V 0.4ns 5V 14.6ns 5V 0.4ns 0V                   7
Vin 3 0 PULSE (0 5 0n 0.4n 0.4n 14.6n 30n)                        8
*Inverter Circuit                                                 9
M1 4 3 0 2 NENH L=2u W=4u AD=32p                                  10
M2 1 4 4 2 NDEP L=4u W=2u AS=32p                                  11
Cout 4 0 0.1pf                                                    12
*Vout 4 0                                                         13
*Include statement to obtain MOS model file                       14
.INCLUDE "full_path_to_spice_model/nmos.3"                        15
*                                                                 16
*For Voltage Transfer Characteristic (VTC)                        17
.DC Vin 0 5 0.1                                                   18
.PROBE DC V(4)                                                    19
*For propagation delay and power                                  20
.TRAN 0.1ns 60ns                                                  21
*For propagation delay                                            22
.PROBE TRAN V(3) V(4)                                             23
*For average power over one full Vin cycle                        24
.MEAS TRAN avgpow AVG POWER FROM=30n TO=60n                       25
.OPTIONS PROBE POST MEASOUT                                       26
.END                                                              27
```

The first line of the SPICE file contains the name of the circuit. This line must always be present. As in the second line, all lines beginning with an asterisk (*) are comments and are optional. This file is more heavily commented than usual.

Lines 4 and 5 specify power supplies which are independent voltage sources. A voltage source identifier always begins with `V` or `v`. In order to interconnect components, nodes represented by integers are used. For example, `Vdd` is connected with its + terminal on node `1` and its – terminal on node `0`. Note that `0` is always used as the ground node. `Vdd` is identified as a `DC` source with value `+5V`. The `+` and `V` are optional. This circuit is somewhat unusual in that it has a substrate bias voltage `Vbb` ≠ `0` to give different circuit characteristics.

`Vin` in lines 7 and 8 is the input voltage to the circuit for transient analyses. Two alternative specifications, PWL (Piece-Wise Linear) and PULSE are given. The PWL specification begins with *; it is commented out and thus is not used although these two specifications are identical for the first 30 ns. PULSE is useful for simple periodic waveforms such as clocks and PWL is useful for non-periodic waveforms. See the on-line SPICE manual for the format of these waveform specifications.

Lines 10 and 11 describe the two MOSFETs in the inverter. The use of `M#` as the identifier designates a MOSFET. The order of the nodes is drain(D), gate(G), source(S) and substrate(B). For example, node 4 is connected to the drain of `M1` and to the gate and source of `M2`. Next, `NENH` and `NDEP` give the name of the model for each of the transistors. The length `L` and width `W` of the channel of each device is given. The abbreviation `u` designates “micro” which, for length in the MKS system, gives microns, μ. In addition, for transient analysis, the areas of the drain of M1 and the source of M2 are given. The abbreviation p designates "pico" which is μ^2. Line 12 specifies a capacitive load on the inverter output of 0.1 pf. Line 15 includes the file `nmos.3` containing the models

for the MOS transistors in this file. "*full_path_to_spice_model*" is the absolute Unix path to the location where you place a copy of the spice model `nmos.3`. Note that *~your_name* as a part of this path will not work. The contents of this file appear later in this section.

The remaining lines specify analyses and outputs. Line 18 performs a DC analysis that produces the voltage transfer characteristic for the circuit. The voltage Vin sweeps from 0 volts to 5 volts in 0.1 volt increments. Line 19 along with the word PROBE on line 26 specifies the output to be V(4). Line 21 performs a transient analysis that will permit the propagate delays, rise and fall times, and average power to be measured. This command begins with `.TRAN` and specifies an output every 0.1 ns for 60 ns. Line 23 along with the word `PROBE` in line 26 specifies that the only outputs are to be the voltages on nodes 3 and 4 which correspond to the input and output of the inverter. Otherwise, HSPICE outputs all voltages plus the power supply currents.

Line 25 uses `.MEAS` (which is the same as `.MEASURE`) to find a scalar value for the average power over a single full period of the `Vin` waveform. The key word `POWER` causes the instantaneous power dissipated by the circuit as a function of time to be calculated. The measurement function `AVG` (average) then takes the average of `POWER` over the time interval specified by `FROM` and `TO` and assigns the value to the output variable `avgpow`.

In general, `PROBE` in a `.OPTIONS` statement as on line 26 restricts the outputs to those specified in `.PROBE`, `.PRINT`, `.PLOT` and `.GRAPH` statements. `POST` in `.OPTIONS` causes the output files to be prepared for AvanWaves. Finally, `.END` designates the end of the SPICE file.

The model file `nmos.3` follows.

```
*Model for Enhancement-Mode MOSFET                                    1
.MODEL NENH NMOS LEVEL=3 RSH=0 TOX=300E-10 LD=0.21E-6                 2
+ XJ=0.3E-6 VMAX=15E4 ETA=0.18 GAMMA=0.4 KAPPA=0.5                    3
+ NSUB=35E14 UO=700 THETA=0.095 VTO=0.781 CGSO=2.8E-10               4
+ CGDO=2.8E-10 CJ=5.75E-5 CJSW=2.48E-10 PB=0.7 MJ=0.5                5
+ MJSW=0.3 NFS=1E10                                                  6
*Model for Depletion-Mode MOSFET                                     7
.MODEL NDEP NMOS LEVEL=3 RSH=0 TOX=300E-10 LD=0.21E-6                8
+ XJ=0.3E-6 VMAX=15E4 ETA=0.18 GAMMA=0.4 KAPPA=0.5                    9
+ NSUB=35E14 UO=700 THETA=0.035 VTO=-2.231 CGSO=2.8E-10             10
+ CGDO=2.8E-10 CJ=5.75E-5 CJSW=2.48E-10 PB=0.7 MJ=0.5               11
+ MJSW=0.3 NFS=1E10                                                 12
```

Lines 2 through 6 and 8 through 12 describe the models for the enhancement-mode and depletion-mode transistors, respectively. Each model begins with `.MODEL` followed by the name of the model, the transistor type, and the HSPICE model parameters for a Level 3 model. Note the + at the beginning of each subsequent line of the model parameter list. The + designates the continuation of a line. Thus, in HSPICE, each model is a single statement.

Executing HSPICE

HPSPICE places almost all of the output files from a run in the run directory, i.e., the directory from which it is executed. So you should change to the directory in which you

would like the result files to appear before executing AvanWaves or HSPICE. If you are analyzing a new circuit `design`, use your favorite editor to enter the initial version of `design.sp`. Then start AvanWaves by entering:

```
awaves
```

To open a design, click **Design:Open**. In the **Open** text area, enter the design file path name if necessary. Then select `design.sp` and click **OK**. If you have not run HSPICE on `design.sp`, you will get a message indicating that you should use **Run HSPICE**. Next, click **Tools:Run HSPICE**. The first time you use the **Run Manager**, click on **Preferences** and select the editor you wish to use. Depending on the editor you use, you may also need to specify `xterm` in the **Xterm command**. Ordinarily, **Machine** and **Version** will be left blank. Click on **OK** to return to the **Run Manager**.

Select the source file you wish to run on the **Design** list and click on **Run**. If you wish to interrupt the run, click on **Stop**. To restart the run, click on **Run**. The **Status** area will give the status of the run. ***Warning:*** *: If design.lis already exists, then the run will end with an* ***Error*** *in the Status area and will not write any output files. You must remove the* `design.lis` *file after each run.* In a separate Xterm window, enter `rm design.lis` and then, before each run on `design.sp`, enter `!r` to remove the `design.lis` file. When the run completes, you can examine `design.lis` by clicking on **Listing**. This file provides a lot of useful information about the parameter values for the run and also indicates errors. Finally, to modify the source file `design.sp` with your editor, click on **Source**.

The HSPICE run will place the various results in files named `design.*` where * represents the file extensions given in Table 1. The `*.lis` file will appear in `design.ext`. # in Table 1 is a sweep number or hardcopy file number.

Displaying HSPICE Output

AvanWaves is used to provide a high-resolution interactive display of HSPICE outputs. It is most easily executed from the directory containing the relevant HSPICE output files. If you are not already in AvanWaves, enter `awaves`. You can access the various result files by clicking on **Design: Open.** ***Warning:*** *Each time you run HSPICE, you must click* ***Panel:Update*** *to display the new results!* The **Results Browser** window contains various output files for your circuit. Clicking on a particular file in the listing and variable type in the **Types** area causes the corresponding variable names to appear in the **Curves** area. To display a particular curve, select it with the left mouse button and drag it with the middle mouse button to the **Panel** in the **Results Display** window. Repeat the select and drag for each curve you wish to display.

If there are curves from a number of sweeps for a given result variable, it is possible to highlight specific sweeps by double clicking on the waveform name in the **Wave List**. This will cause the names of the individual sweeps to appear. Double clicking on a specific sweep name will highlight that sweep. If you want to display only selected sweeps, you can click using the right button in the waveform display area to obtain the context-sensitive menu. Click on **Sweep Filter** and click or drag on those sweeps you want to display**.** Then click on **OK** or **Apply.** These features are illustrated in the AvanWaves tutorial.

TABLE 1. File Extensions

These files appear unconditionally:	
`.lis` or user-specified `.ext`	Output listing including print and ASCII plot outputs
`.st#`	Output status
This file appears if there are subcircuits in the circuit:	
`.pa#`	Subcircuit cross-listing (if there are subcircuits)
These files appear in response to corresponding `.MEASURE` statements in `design.sp`:	
`.mt#`	Transient analysis measurement results
`.ms#`	DC analysis measurement results
`.ma#`	AC analysis measurement results
These files appear in response to `.OPTIONS POST` and corresponding analyses in `design.sp`:	
`.tr#`	Transient analysis results for AvanWaves
`.sw#`	DC analysis results for AvanWaves
`.ac#`	AC analysis results for AvanWaves
This file appears in response to a .SAVE in `design.sp`:	
`.ic`	Operating point node voltages (initial conditions)

Zooming on the waveform display area can be achieved in three different ways: 1) by clicking on **Zoom** in the context-sensitive menu, 2)by clicking on the shortcuts menu bar or 3) by clicking on **Window: Zoom.**

Various measurements can be made on or between waveforms by clicking on **Measure**. For example, suppose you want to measure the propagation delay t_{PHL} for an inverter with input voltage V(3) and output voltage V(4) where the 50% voltage value on the waveforms is at 2.5 V. First, it is necessary to make sure that the right data appears in the resulting measurement label. To set up the data that appears, click on **Measure:Label Options**. Then select only the option **Delta X** and deselect all others since the X axis of the display is Time. Then click on **OK**. Next, select the two waveforms V(3) and V(4) in **Wave List**. To automate the measurement at the 2.5 V level, click on **Measure:Measure Preferences** and select **Lock Horizontal at Y Value** and enter `2.5`. Then click on **Apply**. To actually perform the measurement, click on the **PointtoPoint** shortcut or on **Measure:PointtoPoint. T**hen use the left button to select V(3) and drag to V(4). The measurement points will automatically appear at 2.5 V on V(3) and V(4) and a label appears containing **Delta X**, the propagation delay value.

Output Printing

Plain ASCII outputs such as `design.lis` can be printed using the normal print commands. We advise you to edit out irrelevant text first.

To print from AvanWaves, click on the print icon in the **Shortcut** menu and make the appropriate selections in the **Print** menus. You can print to a printer or send postscript to a file. *Warning: Use only Portrait, not Landscape, to print to a file. Otherwise, the result will be too wide to be viewed or printed.*

To print plots from HSPICE directly, you must use the `.GRAPH` command. See the on-line manual for details.

HSPICE User's Manual and AvanWaves Help and Manual

HSPICE is a very complex and sophisticated circuit analysis tool. As a consequence, we have only touched upon a very small portion of its features here. A three-volume *HSPICE User's Manual* is available on-line for your use. See:

```
/afs/engr.wisc.edu/apps/star-hspice.1998.4/98.4/docs/hspiceManual.pdf
```

The various table of contents, indexes and chapters can be viewed with Frameviewer by entering `viewer`. The table of contents (TOC) and indexes (IX) are hypertext, so you can click on entries to go to the content location. Also, Frameviewer has a **Find** command which allows you to search for keywords. Alternatively, a postscript version is in `Postscript` instead of `Publish`.

AvanWaves provides an on-line help capability. There is also a manual available in postscript form:

```
/afs/engr.wisc.edu/apps/star-hspice.1998.4/98.4/docs/AvanWavesUG.ps
```

In case of persistent problems, you might want to see if you have encountered a known bug by consulting the release notes in:

```
/afs/engr.wisc.edu/apps/star-hspice.1998.4/98.4/docs/hspice_rn98.4.pdf
```

or other documents in:

```
/afs/engr.wisc.edu/apps/star-hspice.1998.4/98.4/docs/
```

HSPICE COMMANDS

In this section, we briefly describe some of the HSPICE commands. The commands may be written using either upper or lower case.

.ALTER

`.ALTER` allows statements in the file to be replaced and a new run to be performed with the new statements. See the manual for more details.

.DATA

`.DATA` allows repeated runs to be performed with the variable values given in a data table. See the manual for more details.

.DC

`.DC` performs a DC analysis. For example `.DC Vin 0 5 0.1` performs a DC analysis using a sweep of Vin from 0 V to 5 V in 0.1 V increments.

.END

`.END` is required at the end of an HSPICE netlist file.

.GLOBAL

.GLOBAL declares a variable to be the same whether in the main circuit or any subcircuit. For example, `.GLOBAL Vdd` will cause all nodes throughout the netlist file labeled Vdd to be connected together. This avoids extra nodes in defining subcircuits.

.GRAPH

`.GRAPH` can be used to plot output directly from HSPICE. Plots from all `.GRAPH` statements in a `design.sp` file appear in a single postscript file. Note that `.GRAPH` is capable of handling a sweep of only one variable. If a second variable is swept, each of its values produces a separate plot. The outputs of `.GRAPH` are `design.gr#`, where # is an integer.

.IC

`.IC` is useful for setting initial conditions for transient solutions. This is essential in whenever the circuit stores information, such as in latches, flip-flops or on dynamic storage nodes. For example, `.IC V(1)=5 V(2)=0` initializes node 1 to 5 V and node 2 to 0V. To use for initialization of a transient solution, include `UIC` in the `.TRAN` statement. A similar command useful for DC solutions is `.NODESET`.

.INCLUDE

Used to include text from one file in another at run time. For example,

```
.INCLUDE '/pong/usr5/k/kime/public_html/555/spice_models/scn06hp.L3'
```

is used to include the text in the file `scn06hp.L3` for the level 3 models for the CMOS devices in an HP 0.6 µ process. Note that the abbreviated path `~kime` will not work.

.LIB

`.LIB` is used to include text from one file in another at run time. It is typically used for the models for elements such as MOS and BJT transistors and transmission lines. Multiple models may appear in the same library file. Thus, the model name must be specified after the file path/filename for the library. For example,

```
.LIB '/pong/usr5/k/kime/public_html/555/spice_models/scn06hp.L13' NOM
```

is used to include the models in library `NOM` in file `scn06hp.L13` which are the nominal, level 13 (BSIM) models for the CMOS devices in an HP 0.6 µ process. Note that the abbreviated path `~kime` will not work.

.MEASURE

`.MEASURE` (abbreviated `.MEAS`) is used to make a variety of measurements on result variables. `.MEAS TRAN avg_id AVG I1(M2) FROM=30n TO=60n` assigns the average value of the drain current of transistor M2 over the interval from t = 30 ns to t = 60 ns to the variable *avg_id*. The HSPICE manual should be consulted for the details of the many other uses of `.MEASURE`. To produce files containing measurement results, `.OPTIONS` should include `MEASOUT`.

.MODEL

`.MODEL` files define the models for elements such as MOS and BJT transistors and transmission lines.

.OPTIONS

`.OPTIONS` is followed by a list of parameters, in some cases, accompanied by parameter values. It is the main mechanism for controlling the simulation including outputs and tolerances. See the manual for details.

.PARAM

`.PARAM` is used to define a parameter that is used as a variable in other statements in the netlist file. For example, suppose you want to run a circuit with different sized devices that are scaled using parameter λ. Then `.PARAM lw = 0.3u` can be used for

a 0.6 µ process and `.PARAM lw = 0.2u` can be used for a 0.4 µ process. The device dimensions can be specified in terms of `lw: L='2*lw' W='3*lw' AD='18*lw*lw' AS='18*lw*lw'` For continuation to the next line within the single quotes `'`, use \\ and outside of the quotes `'`, use \.

.PLOT

`.PLOT V(2) V(3)` plots selected output variables, in this case V(2) and V(3), to `design.lis` using ASCII characters. `.PLOT` is useful for looking at plotted results without access to AvanWaves. It can also be used with `.OPTIONS PROBE` to select the values to be placed in output files.

.PRINT

`.PRINT V(2) V(3)` produces a printed table of node voltage values for nodes 2 and 3 in the `design.lis` file. It can also be used with `.OPTIONS PROBE` to select the values to be placed in output files.

.PROBE

`.PROBE` is used with `.OPTIONS PROBE` to select the values to be placed in output files. Unlike `.PLOT` and `.PRINT`, it does not place results in `design.lis`.

.TRAN

`.TRAN` performs a transient analysis. For example, `.TRAN 0.1ns 50ns` provides 501 points of transient analysis output at 0.1 ns intervals from t = 0 through t = 50 ns. To produce useful output, `.TRAN` requires a waveform on at least one input.

.WIDTH OUT=80

This command sets the number of columns to 80 for use with a terminal or 80-column output device.

INPUT WAVEFORMS

PWL and PULSE are two input waveforms frequently used in transient analyses. The format for a piece-wise linear (PWL) waveform is:

```
PWL time0 value0 time1 value1 time2 value2 ...
```

where, at time*i*, the value of the waveform is set to value*i* and remains there until value(*i*+1). The value set at the last time remains until t = ∞. For example,

```
Vin 3 0 PWL 0ns 0V 0.4ns 5V 14.6ns 5V 0.4ns 0V
```

starts *Vin* at 0 at *t* = 0, ramps it to 5 volts by *t* = 0.4 ns, and holds it there until*t* = 15ns. It then ramps *Vin* down to 0 volts by *t* = 15.4 ns, and holds it as 0 volts, thereafter.

The format for a PULSE waveform is:

```
PULSE (intial_value pulse_value delay risetime falltime
       duration period)
```

The waveform begin at the `intial_value`. The `delay` is the time from t = 0 to the beginning of the pulse. The `risetime` is the time required to go from the

intial_value to the pulse_value. The falltime is the time to go from the pulse_value to the intial_value. The duration is the time that the pulse remains at the pulse_value. The period is the length of time from the end of delay to the rising edge of the next pulse. For example,

```
Vin 3 0 PULSE (0 5 4.9n 0.2n 0.2n 4.8n 10n)
```

represents a pulse waveform that begins at 0, has a pulse beginning at t = 5 ns measured at the 50% point with rise and fall times of 0.2 ns and a duration measured at the 50% value of 5 ns. It repeats with a period of 10 ns.

OUTPUT VARIABLES

In .PROBE, .PRINT, .PLOT and .MEASURE statements, voltages, currents, power and user-defined variables appear. The general form for voltages is V(n1<,n2>) where < > indicates optional. If ,n2 is missing then V(n1) is simply the voltage on node *n*1 referenced to GND. If ,n2 is present, then V(n1,n2) is the voltage on node *n*1 with reference to node *n*2, $V(n1) - V(n2)$.

For a current through a voltage source, the simplest form is I(Vxxx) where Vxxx is the voltage source name. For example, the current from the power supply Vdd is the negative of I(Vdd). To measure specific currents, it is sometimes useful to add a 0-volt source Vx to generate a new node. You can then output current I(Vx). For example, this is useful for obtaining the current for calculating the power dissipation for a CMOS NAND gate. In this case, Vx has its positive terminal attached to Vdd and negative terminal attached to the sources of the PFETs in the NAND.

For the current through a branch of an element, the simplest form is: In(Wwww) where n is the node position number of the branch on the element and Wwww is the element name. For example, the current into the drain of n-channel MOSFET M1 is I1(M1) and the current out of its source is I3(M1) since the terminals on a MOSFET are in order: 1) drain, 2) gate, 3) source, and 4) substrate.

The instantaneous power dissipated in an element is specified as P(Wwww) where Wwww is the element name. For example, the power dissipated in an MOSFET M1 is P(M1). The instantaneous power dissipated in the entire circuit is denoted by key word POWER.

Since average power in MOS circuits has a strong transient component, determination of average power requires a clear picture of current flow in the circuit in order to select a correct time interval for the measurement. Also, to conveniently measure the power for a portion of the circuit, a separate power supply source or a 0V voltage source attached to the main power supply source may be needed to supply power to the portion of the circuit.

EXPRESSIONS

Algebraic expressions are of use in obtaining a number of results frequently used in device and circuit analysis. In Table 2, we will summarize selected operators and functions useful in analyzing devices and digital electronic circuits. We will particularly focus on the use of expressions in analyzing and making measurements on outputs.

TABLE 2. Frequently-used Operators and Functions in AvanWaves and HSPICE

Operation/Function	Notation
Add	+
Subtract	–
Multiply	*
Divide	/
x to the power y	pow(x,y)
Square root of x	sqrt(x)
Absolute value of x	abs(x)
Derivative of x	derivative(x)
Integral of curve	integral(curve)*

* Available in AvanWaves and `.MEASURE` only in HSPICE.

AvanWaves also has comparison, Boolean and conditional operators. `.MEASURE` in HSPICE has functions such as average, peak-to-peak, and RMS.

Example: Find the power consumption during a period of an applied input pulse wave of an NMOS inverter with power supply voltage`Vdd` and depletion mode transistor M2. The duration of the transient analysis and the pulse period is 30 ns. In AvanWaves, we can use the **Expression Builder** to construct an expression `power` equal to:

```
integral(I1(M2))*5.0/60n
```

In terms of the processing of outputs, expressions such as that above can be used within HSPICE in .PLOT, .PRINT, .PROBE and .MEASURE statements as well as being constructed in **Expression Builder**.

Expression Builder can be opened by clicking the **Expression Builder** icon on the shortcut menu or by clicking **Tools:Expressions**. To build the equation above, double click on `integral` in the **Functions** area to add `integral` to the **Expression** field. Change the **Result Browser** type to Current1 and drag the current `I1(M1)` using the middle mouse button into the parentheses for `integral`. Next drag `*` and `/` from the **Operators** area and enter 5.0. Finally, drag / from the **Operators** area to the **Expression** and enter 60*n*. In the Result field, enter `power` and then select **Apply** to cause `power` to appear in the **Expressions** area. Finally, select `power` and, with the middle mouse button, drag it to a panel. Note that `power`, by definition, is a horizontal line. Note that the various functions can be applied only to single variables, not to expressions. If you wish to apply a function to an expression, define the expression and give it a name and click **Apply** so that it appears in the **Expressions** area.

In order to form power within HSPICE, we use .MEASURE since there is no average or integral function for use outside of .MEASURE. Using the AVG function,

```
.MEASURE power AVG POWER FROM=30n TO=60n
```

To avoid effects of the initial conditions on the power calculation, a full cycle of the input Vin precedes the beginning of the measurement.

SPECIFIC ANALYSES AND OUTPUT

Examples are given for performing specific frequently-used analyses. The HSPICE files and results are available in www.cae.wisc.edu/~kime/555/tool_docs/spice_examples. To go directly to these files on CAE Unix workstations, replace www.cae.wisc.edu/~kime with ~kime/public_html. The HSPICE circuit for the I-V Curve examples is:

```
NMOS I-V Curve Circuit (file: nmos_iv.sp)
*Last revised 1/11/97
*The voltage values given are nominal values to be used
*if not over-ridden by dc or transient commands
*affecting the source value.
Vgs 1 0 2V
Vds 2 0 2V
Vbs 3 0 0V
*NFET
M1 2 1 0 3 NENH L=3u W=3u
*NFET Model
.MODEL NENH NMOS LEVEL=3 RSH=0 TOX=300E-10 LD=0.21E-6
+ XJ=0.3E-6 VMAX=15E4 ETA=0.18 GAMMA=0.4 KAPPA=0.5
+ NSUB=35E14 UO=700 THETA=0.095 VTO=0.781 CGSO=2.8E-10
+ CGDO=2.8E-10 CJ=5.75E-5 CJSW=2.48E-10 PB=0.7 MJ=0.5
+ MJSW=0.3 NFS=1E10
*
.options post probe
*For I-V Curve Id vs Vgs
.dc Vgs 0 5 0.1
.probe I1(M1)
*For I-V Curve Id vs Vds with Vgs = 2V
.dc vds 0 5 0.1
.probe I1(M1)
*For family of I-V Curves
.dc Vds 0 5 0.1 Vgs 0 5 1
.probe I1(M1)
.end
```

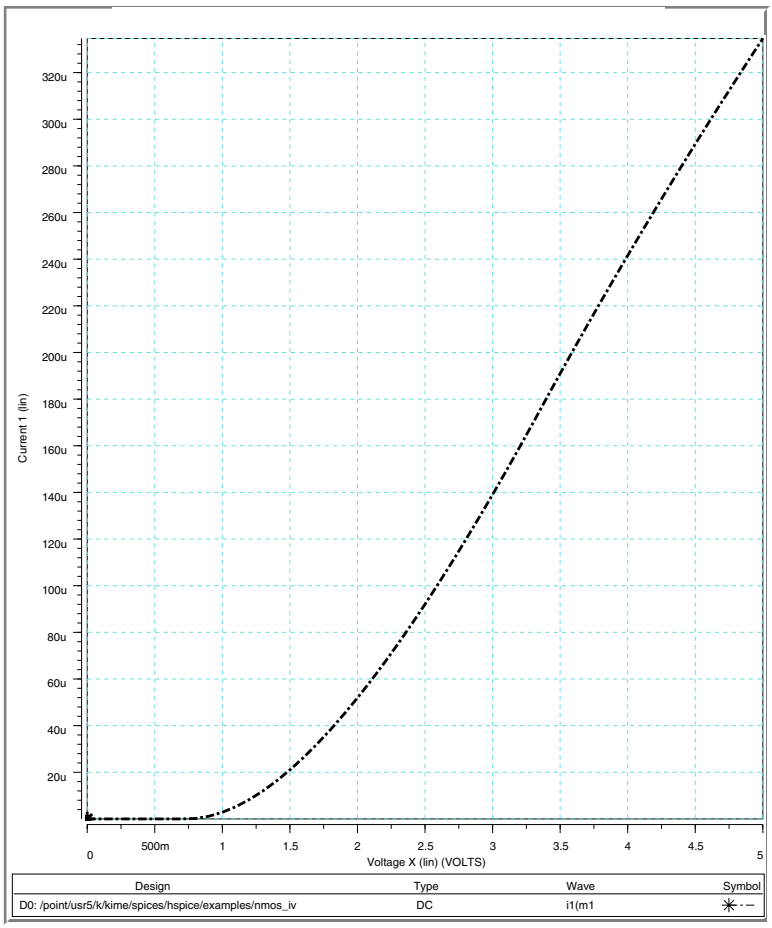
I-V Curves

For the above circuit, to obtain the data to plot an I-V curve using AvanWaves: I_D versus V_{GS}, we use:

```
.dc Vgs 0 5 0.1
```

```
.probe I1(.probe I1(M1)
```

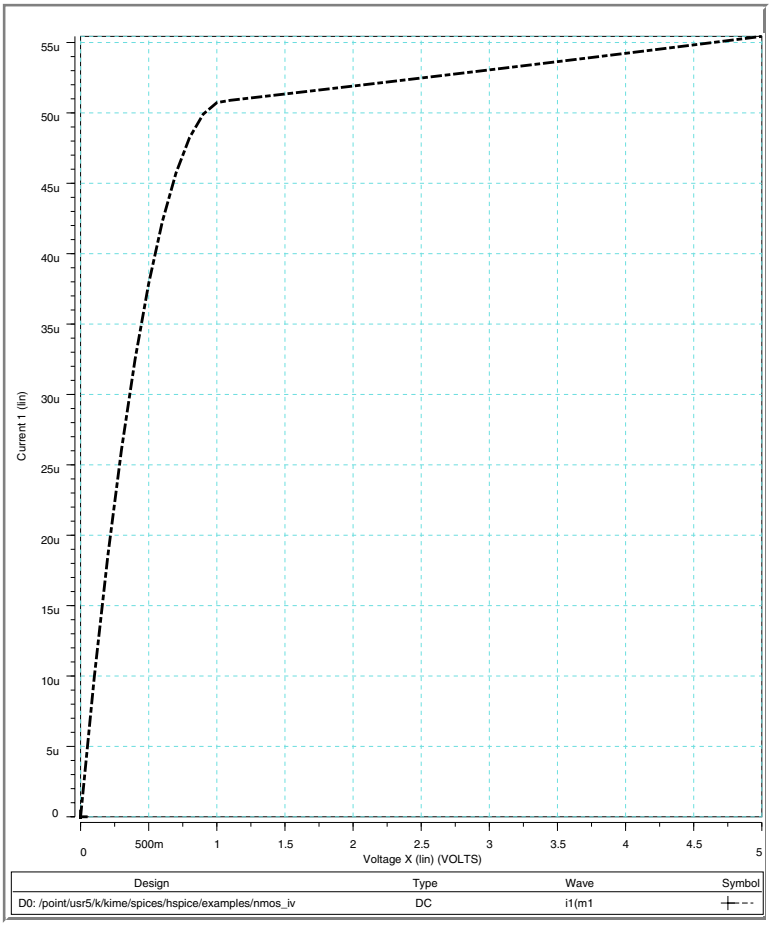
Vgs sweeps from 0 to 5 volts in steps of 0.1 volt. The output current into the drain of M1 is I_D. V_{DS} for this analysis is the nominal value of 2V given in the HSPICE file. The resulting curve is:



To obtain the I-V curve: I_D versus V_{DS}, we use:

```
.dc vds 0 5 0.1
.probe I1(M1)
```

`Vds` sweeps from 0 to 5 volts in steps of 0.1 volt. The output will be V(2) which is V_{DS} and the output current into the drain of M1which is I_D. V_{GS} for this plot is the nominal value of 2V given in the SPICE file. The resulting curve is:



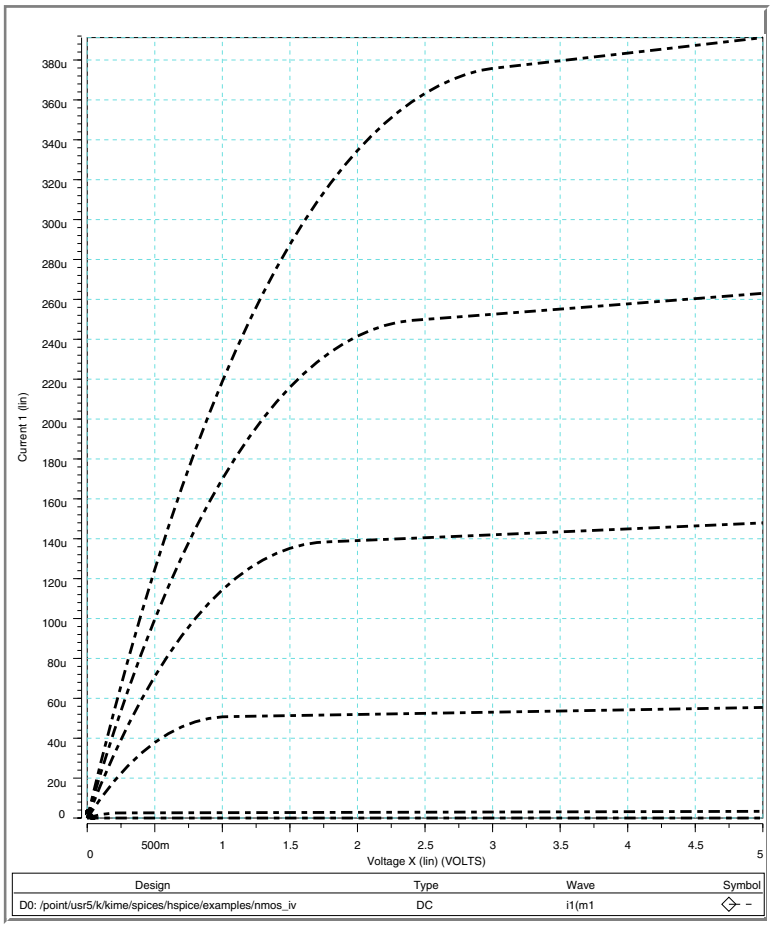
In general, for both dc and transient analyses, one most be careful not to request too many or too few result points. In this case, there are about (5 – 0) / 0.1 = 50 result points, which is reasonable. To specify `tran 0.1ns 100ms`, which requests `100ms/0.1ns` or about 1 billion result points, clearly is not! If a resolution of 0.1ns is required, then the simulation time should be restricted to no more than 100 ns even for simple circuits.

Family of I-V Curves

To obtain the family of I-V curves I_D versus V_{DS} with each curve for a value of V_{GS}, we use:

```
.dc Vds 0 5 0.1 Vgs 0 5 1
.probe I1(M1)
```

This sweeps `Vds` from 0 to 5 volts in steps of 0.1 volt for every value of `Vgs` from 0 to 5 volts in steps of 1 volt. The result is a family of six curves, one for each value of `vgs`:



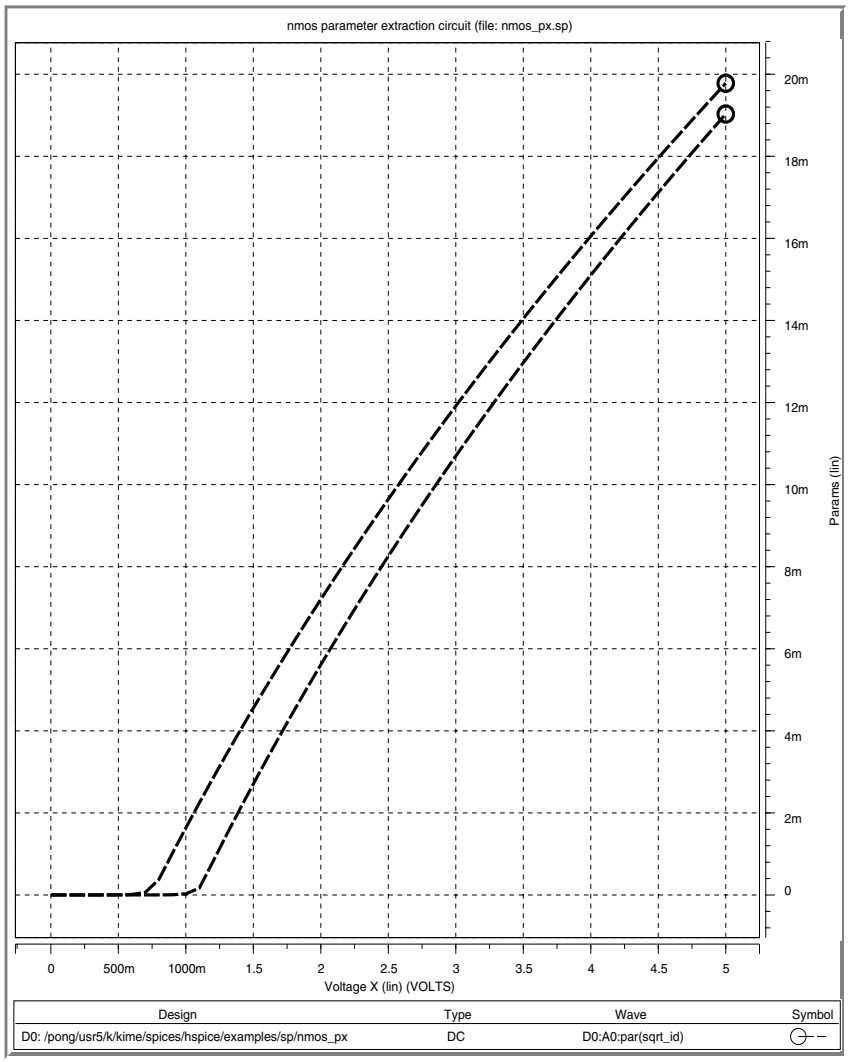
I-V Curve for Parameter Extraction

This gives a plot of $\sqrt{I_D}$ versus $V_{DS} = V_{GS}$ as shown on pages 69-70 of Kang and Leblebici. I_D is again I1(M1). The SPICE circuit used is:

```
NMOS Parameter Extraction Circuit (file: NMOS_PX)
*Last revised 1/11/97
*Vds = Vgs
Vgs 1 0 0V
Vsb 0 2 0V
*NFET
```

```
M1  1  1  0  2 NENH L=3u W=3u
*NFET Model
.MODEL NENH NMOS LEVEL=3 RSH=0 TOX=300E-10 LD=0.21E-6
+ XJ=0.3E-6 VMAX=15E4 ETA=0.18 GAMMA=0.4 KAPPA=0.5
+ NSUB=35E14 UO=700 THETA=0.095 VTO=0.781 CGSO=2.8E-10
+ CGDO=2.8E-10 CJ=5.75E-5 CJSW=2.48E-10 PB=0.7 MJ=0.5
+ MJSW=0.3 NFS=1E10
*
.options post probe
*For parameter extraction
.dc Vgs 0 5 0.1 vsb 0 3 3
.probe sqrt_id=PAR('sqrt(I1(M1))')
.end
```

The resulting I-V curves are useful for finding k_n, V_{TO}, and *GAMMA*.

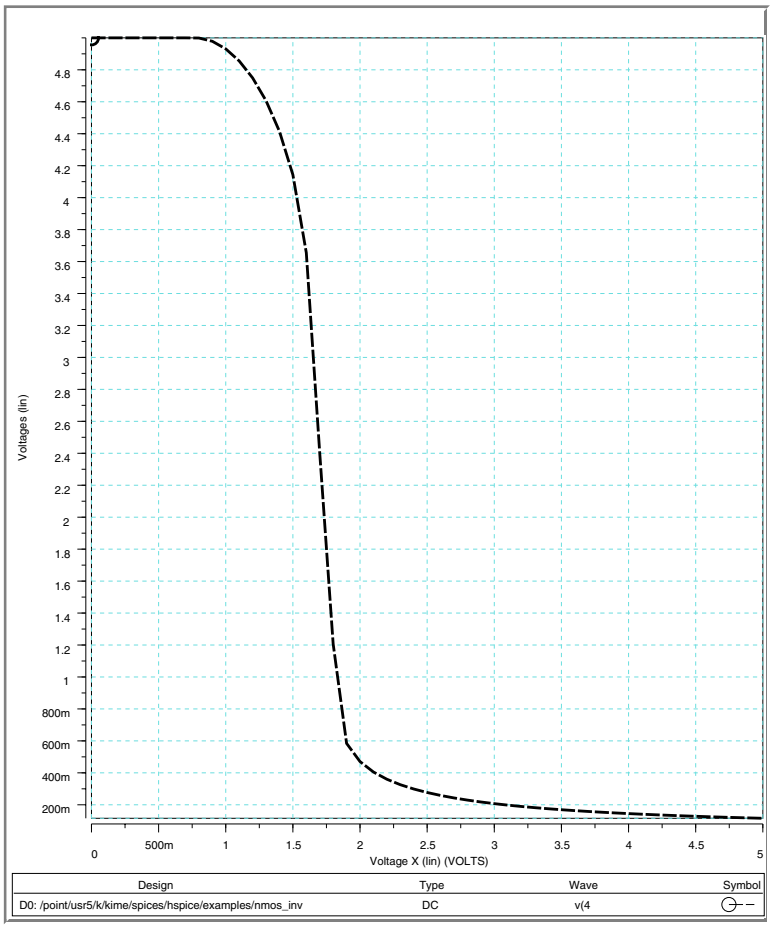


Voltage Transfer Characteristic (VTC)

For the NMOS inverter example, the node for the input voltage `Vin` is 3 and the node for the output voltage is 4. To obtain the VTC, we used:

```
.dc vin 0V 5V 0.1V
.probe dc V(4)
```

This command replaces the waveform for Vin specified in the file with a signal that sweeps from 0 to 5V in increments of 0.1V. For the example circuit, the VTC obtained is:



Transient (Delay) Analysis

For the NMOS inverter example, the node number for the input voltage is 3 and the node number for the output voltage is 4. In this case, the input voltage waveform `Vin` to be used,

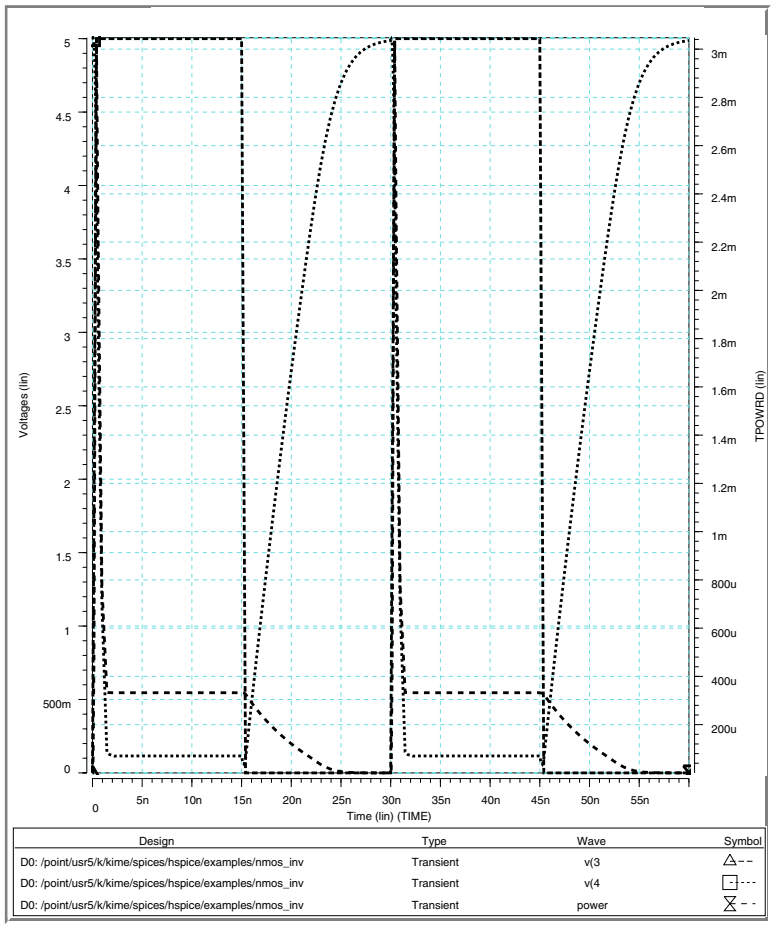
```
Vin 3 0 PULSE (0 5 0N 0.4N 0.4N 9.6N 20N)
```

is in the circuit file. Look at independent voltage sources to see the meaning of the various numbers specifying the piece-wise linear (PWL)or pulse (PULSE) waveform in the manual. To perform the transient analysis, we used:

```
.tran 0.1ns 20ns
.probe tran V(3) V(4)
```

For this command, a simulation value is determined every 0.1 ns for an interval of 20 ns giving 201 result points.

The circuit behavior is analyzed in the time domain by using the voltage waveform, specified for Vin as the input stimulus. In this case, the input waveform is a single pulse of 10 nanoseconds with 0.4 nanosecond rise and fall times. Both the input waveform v(3) and the output waveform v(4) are plotted versus time and can be viewed with AvanWaves. In addition, the curve for the total instantaneous power, POWER, is also plotted.



CONVERGENCE PROBLEMS

Very infrequently, one encounters some form of numerical convergence or other numerical problem when using HSPICE. Due to the low likelihood of these problems, we give references only to the on-line manual.

DC Analysis

See page 5-5 and pages 5-8 through 5-26 of the on-line manual.

Transient Analysis

See pages 6-6 through 6-19 of the on-line manual. Note that DC convergence problems can also affect convergence in transient analysis.

Diode Convergence

For handling of convergence for regular diodes, see page 12-2 of the on-line manual. For handling of convergence for MOSFET diodes, see page 15-19.

EXECUTING HSPICE FROM A COMMAND LINE

HSPICE places almost all of the output files from a run in the directory in the run directory, i.e., the directory from which it is executed. So you should change to the directory in which you would like the result files to appear. Execute HSPICE from a command line by simply entering:

```
hspice design> design.ext
```

This will cause HSPICE to be executed on the netlist file `design.sp` and will place the various result files in files named `design.*` where `*` represents the file extensions given in Table 1. The `*.lis` file will appear in `design.ext`.

Another way to run HSPICE is to simply type `hspice`. You will be prompted for the input and output file names and will be asked to answer a number of questions. Normally you will use the default answers.

REFERENCES

S-M. Kang and Y. Leblebici, *CMOS Digital Integrated Circuits: Analysis and Design*, 2nd Edition, New York: McGraw-Hill, 1999.

G. Massobrio and P. Antognetti, *Semiconductor Device Modeling with SPICE*, 2nd ed., New York: McGraw-Hill, 1993.

Star-Hspice Manual Version 98.4, Avant! Corporation, January 1998.

AvanWaves User Guide, Version 97.2 for use with AvanWaves Release 97.2, Avant! Corporation, May 1997.

FEEDBACK

Please e-mail information on errors, suggested additions or other comments on these notes to C. R. Kime, at `kime@engr.wisc.edu`.